Liver Disease Prediction

Ali Saeidi Ashtiyani, Ragini Dwivedi ,Akshay kumar Gyara and Dimitrie Milinovich

Department of Computer Engineering, Data Mining (CMPE255)

Email: ali.saeidiashtiyani@sjsu.edu, ragini.dwivedi@sjsu.edu, akshaykumar.gyara@sjsu.edu, and dimitrie.milinovich@sjsu.edu

Github Link: https://github.com/alisaeidi92/Liver-Disease-Prediction

Abstract

Numerous patients suffer every year of liver disease. However, without invasive procedures like surgery, it's impossible to verify the presence of disease with certainty. According to Mayo Clinic, "Abnormal liver function test results don't always indicate liver disease." As a result, the inability for medical professionals to deliver confident answers leaves patients with fear, and doctors with an uncertain pathway to treatment. Our goal is to resolve these problems by increasing the accuracy of current tests using data mining and machine learning strategies.

Introduction

As Machine Learning (ML) is getting more advanced and heavily researched, its applications in health and medicines is becoming more popular. Data mining and machine learning can be used to diagnose patients with certain diseases based on their test results.

This project focuses on detecting patients with liver disease based on their Comprehensive Metabolic Panel (CMP) blood work results. We have processed the data and developed multiple ML algorithms that can classify sick patients based on their bloodwork with high accuracy. Moreover, this project provides a great insight and comparison between different methods of data mining and pros and cons of different ML algorithms for medical data.

Backgrounds and Similar Works

There have been countless number of researches and projects on medical datasets, specifically for liver diseases. This project is based on a dataset found on kaggle website named as Liver Disease Lab Data[1]. There have been no previous work or tasks on this specific dataset, however, other datasets such as Indian Liver Patient Records[2] have been heavily researched and multiple solutions were provided. Some works suggested working with CNN algorithms that provide 100% accuracy while some other methods demonstrated a good accuracy using ML models such as SVM or KNN. There have been other works with lower accuracy with Logistic Regression and Naïve Bayes algorithms.

Data and Features

The dataset has 10 columns as features and 1 column as the target class. Features are namely; Age, Total Bilirubin, Direct Bilirubin, Alkaline Phosphotase, Alamine Aminotransferase, Aspartate Aminotransferase, Total Proteins, Albumin and Albumin and Globulin_Ratio which are taken from CMP of each patient. The target class named Liver Disease indicates the presence of liver disease for a patient. Figure 1 shows the unbalanced target class in this dataset.

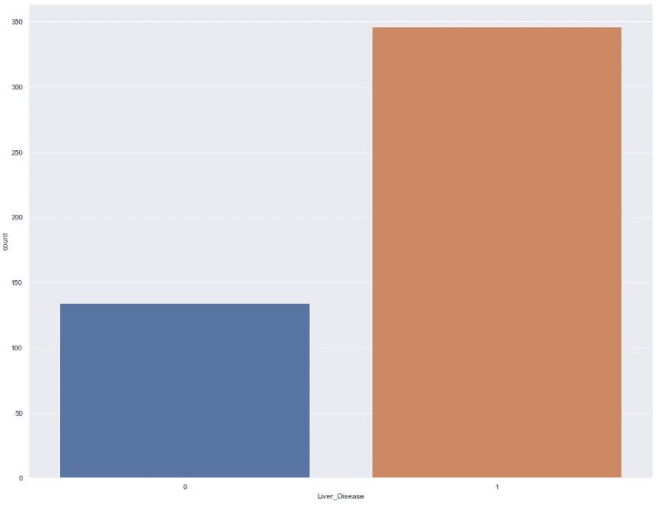


Figure 1 - unbalanced Liver Disease Instances

The imbalance in the dataset will definitely have negative effects in learning of ML algorithms from the dataset. Also, Figure 2 shows the imbalance in the Gender and the fact that men are more likely to be sick than women.

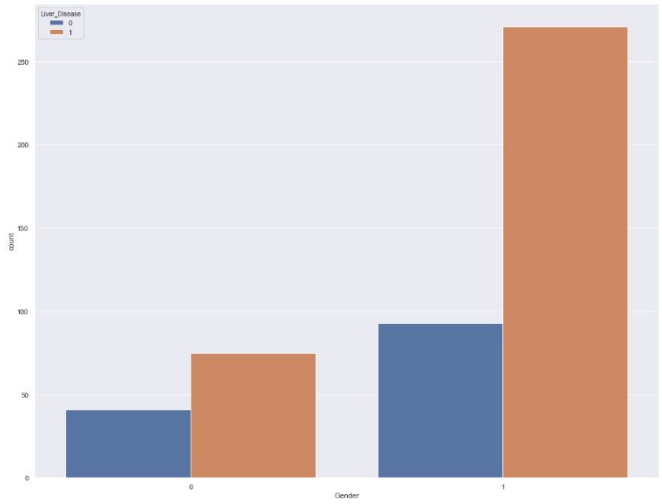


Figure 2 - Imbalance between Male and Female

Processing the Data

In this project a good number of pre-processing methods have been developed and used for some algorithms. Since different ML algorithms were used in this project, different processing had to be done to tune the parameters and reach the ideal accuracy. The categorical data such as gender had to be converted to binary format. As the number of rows with missing value were very small, they were removed in this project.

As we can see in Figure 3, some features in our dataset are highly correlated. Such correlation in data may create problems if the dataset is not large enough. Since the dataset used in this project contains only a few hundred instances, the high correlation between columns should be handled. The best method to overcome such issues is to use PCA which is a method for dimensionality conversion and reduction. Therefore, PCA was used to ensure that each feature is being processed in a different dimension. Also, PCA enables us to take in account the most effective features rather than considering all features equally contributing in the results.

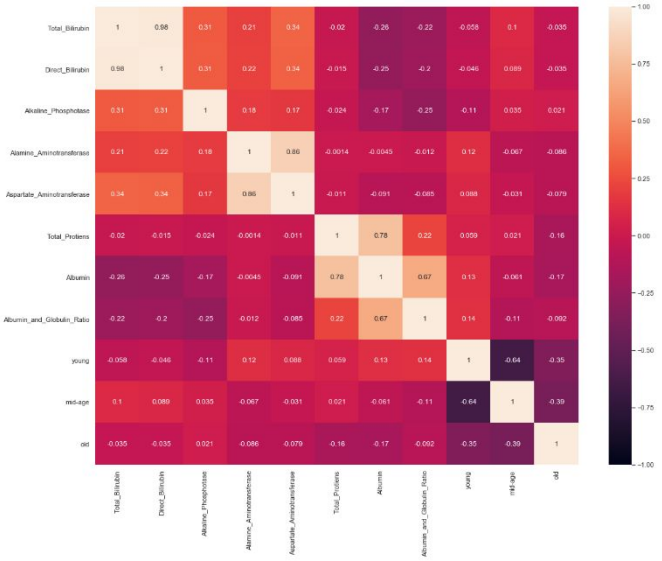


Figure 3 – Correlation Between Features

The outliers in this dataset are tricky to deal with. Methods have been developed in the project to deal with outliers in the data however, they were only used in a few of our approaches. The main reason in some features, outliers actually are indications of liver disease. For example, for some liver conditions a certain enzyme may be elevated to 1000s range while the mean is around just 30. Therefore, to deal with outliers in such instances we set the maximum value to be 1000. In this scenario while we partially eliminate the outliers, the information conveyed by such instances are not lost. Figure 4 shows the existence of outliers in this dataset.

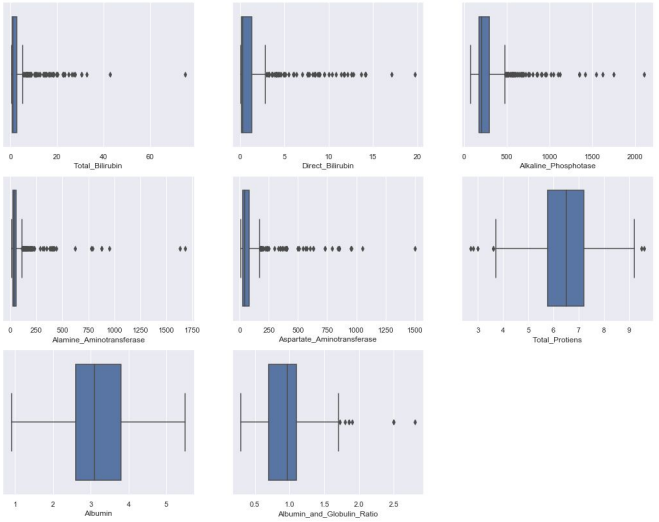
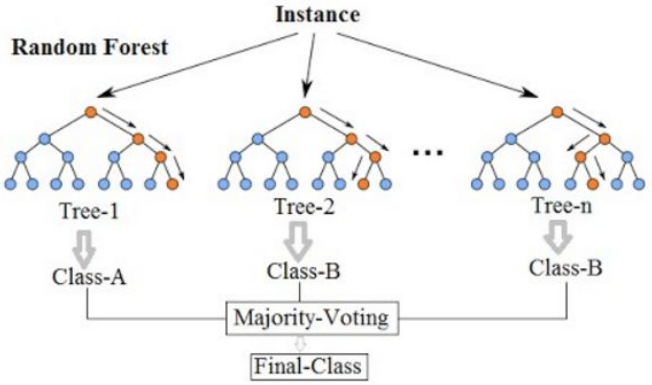


Figure 3 - Outliers in Dataset

Algorithms

Random Forest Classifier

Random Forest classifier (RF) is an accurate and a very fast classifier compared to other classifiers. It uses uncorrelated trees as ensembles, each tree providing a classification result and the highest vote will be chosen as the class for that specific instance. Figure 2[3] shows the idea behind RF classifiers.



The RF classifier built in sklearn is used in this project for classification of liver disease. Even though the sklearn RF classifier has numerous parameters to be set, we chose the most important parameters to tune which are namely; n_estimators, criterion, min_samples and max_features. The n_estimators parameter is the number of ensembles or trees used to classify each instance which is the most important attribute in RF classifier. The best n_estimator is either chosen by trial or other methods such as GridSearchCV.

Support Vector Machine

SVM is a supervised learning computation with related learning algorithms which analyzes data used for regression analysis and classification. SVMs are generally utilized for learning ranking functions, regression or classification. A Support Vector Machine (SVM) isolates the information into two categories of performing grouping and building a N-dimensional hyperplane. SVM depends on statistical learning hypothesis and basic risk minimization rules and has the goal of deciding location of decision boundaries. This is also known as hyperplanes that produce the optimal separation of classes.

An indicator variable which is considered an attribute and a transformed attribute that is utilized to characterize the hyper plane is known as a feature[18]. Here, picking the most appropriate representation can be taken as feature selection. A lot of features that describe one case is known as a vector.

The objective of this modeling is to locate the ideal hyperplane which isolates clusters of vectors in such a manner that cases with one classification of the targetVariable are on one side of the plane and cases with the other classification are on the opposite side of the plane. The vectors closest to the hyperplane are the support vectors[5] as in figure 5[6].

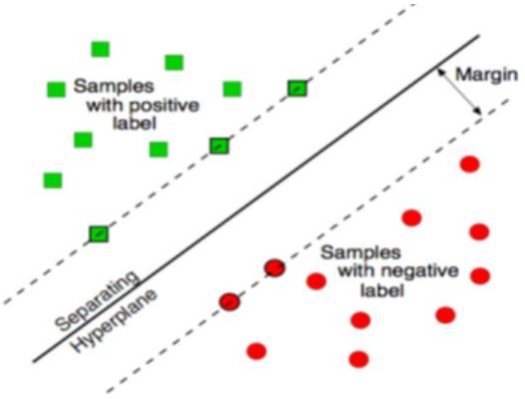


Figure 5 - SVM Classifier (Hyperplane)

K-Nearest Neighbors

The K- Nearest Neighbors (KNN) is a supervised learning algorithm that can be used as either a classifier or reggresor. It is a versatile and robust classifier which is commonly used as a benchmark for other complex classifiers such as “Artificial Neural Networks” (ANN) and “Support Vector Machines” (SVM). This classifier has outperformed other classifiers and has been used in various applications such as genetics, pattern recognition, economic forecasting and recommendation systems.

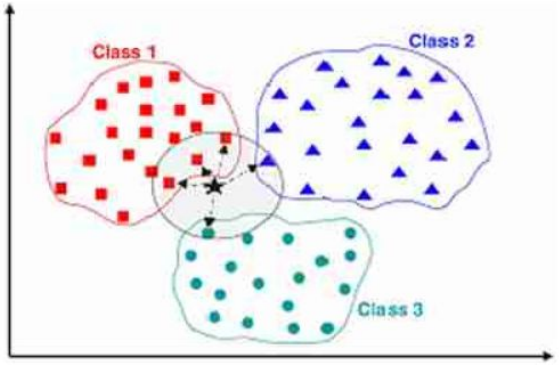


Figure 6 - An Example of KNN Classifier

KNN depends on feature similarity i.e. when given a training data set it tries to find the features which are close to classify a given point . In the above figure there are three classes : class 1, class 2 and class 3. It uses the KNN algorithm to estimate or predict the class of black points by

finding the nearest neighbours to it. More simply this means given a dataset with observations (X,Y) where X is the predictor and Y is the target or class attribute . Our goal is to find a relation between X and Y and define a function f : X implies to Y i.e. given an unseen observation X, f(X) can accurately predict the output Y.

KNN is also known as Lazy Algorithm . As it doesn't have much parameters to play with. The most important parameter here is "n_neighbors = k". We have to choose the value k in such a way that we can get a best fit possible. Mostly we have to choose k as an odd number as it takes the majority votes of nearest neighbors to predict the class label. Ex if we take k = 6 and we get class 1 as 2, class 2 as 2 and class 3 as 2 , it becomes difficult to decide which class label to be used. The best practical values for k ranges from 5 to 10 [4]. We can find the K value using parameter estimators such as gridsearch or randomly using different k values. Other default parameters are 'weight = uniform' where all the points are equally weighted for each neighborhood , 'metric = minkowski' which is used to measure the similarity .The default distance metric used by KNN is Euclidean distance also known as Minkowski distance.

Results

Random Forest Classifier

To get the best results, We used different combinations of pre-processing methods and parameter tuning. The most important metrics to consider in this part was the recall and f1 scores. Recall score indicates the number of sick patients that were detected from all the sick patients which should be the first priority of this project. We used GridSearchCV to find the best parameters for the model. Moreover, oversampling (over) and PCA were used for some methods. Additionally, two standardization methods were used namely; MinMaxScaler and StandardScaler from sklearn library. Below, we can see the table containing the results of applying combinations of these methods and the results.

PCA	Over #	AgeGrp	outliers	scaler	recall	F1
No	Yes	Yes	No	MinMax	0.76	0.78
No	Yes	Yes	Yes	MinMax	0.79	0.81
No	No	No	No	MinMax	0.90	0.84
No	Yes	No	No	MinMax	0.75	0.81
No	Yes	No	Yes	MinMax	0.82	0.80
4	No	No	No	MinMax	0.83	0.82
2	No	No	No	MinMax	0.93	0.85
6	No	No	No	MinMax	0.83	0.83
8	No	No	No	MinMax	0.85	0.84
6	Yes	No	No	MinMax	0.72	0.80
No	No	No	No	Std	0.87	0.85
6	No	Yes	Yes	Std	0.83	0.82
6	Yes	Yes	Yes	Std	0.72	0.80

The recall and F1 score are for the sick patients (target=1). For most methods, there has been a tradeoff between the accuracy on 0 targets and 1 targets meaning when the recall on sick patients is very high, the recall for non-sick patients tends to be too low. Meaning that the model is classifying more patients as sick to be able to capture all the sick patients. However, oversampling and higher PCA value seems to solve this problem to some extent and create a balance between the recall values for sick and un-sick patients.

Support Vector Machine

Before applying SVM, we implemented a naive predictor whose job is to return that every datapoint has *Disease=True* and calculate accuracy, TPR, FPR metric on that predictor.

MinMaxScaler, standardization method was used from sklearn library to preserve zero entries if the feature matrix is sparse. By analyzing the dataset, it was found that different values of features do not have any inherent ordering. To solve this issue One-hot encoding was used.

ROC plot is used to represent TPR(True positive rate) vs FPR(False positive rate). It also represents the performance of the machine learning algorithm. ROC is helpful in our case because simply knowing the number of correct predictions would not be sufficient. Figure 7 and 8 shows the ROC (Receiver Operating Characteristics), accuracy and F-score.

```
For classifier Support Vector Classification:
Unoptimized model
------
Accuracy score on testing data: 0.7579
F-score on testing data: 0.9399

Optimized Model
------
Final accuracy score on the testing data: 0.7579
Final F-score on the testing data: 0.9399
Best parameters:
{'C': 0.001, 'kernel': 'poly'}
For classifier Support Vector Classification, ROC score is 0.500000
```

Figure 7 - Accuracy and F-score for SVM

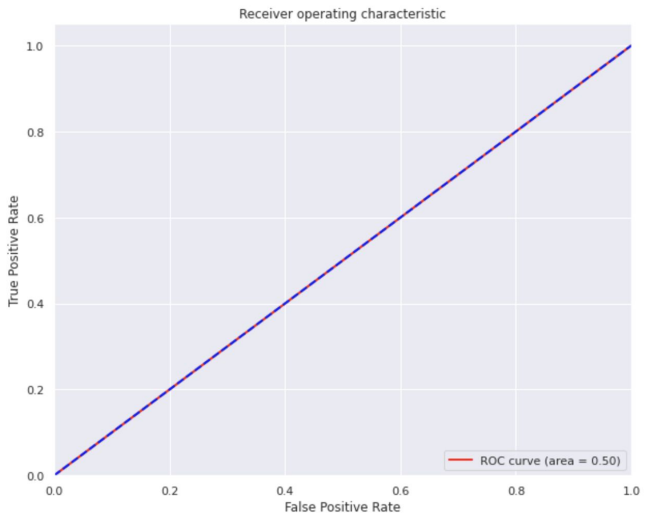


Figure 8 -SVM ROC Curve

According to the data gathered using SVM, it is evident that SVM shows pretty good results in terms of F-beta score which is **0.9399**. However, its AUC(area under the curve) is surprisingly low (**0.5**).

K-Nearest Neighbors

For this model initially we just used a basic preprocessing method to remove null value, then changed the categorical variables (gender) to numerical value , created a training and testing set and then trained the model. For this model we just randomly selected values for k. Then tried to reduce the dimensionality using PCA as it helps to improve the distance metric and performed the same procedure as above. Rescaled the data using StandardScalar() and MinMaxScalar() to normalise the data so it can reduce the number of misclassifications. Performed oversampling to balance out the no of '1' and '0' class labels. (1= 347, 0 = 136). Used different distance metrics such as Euclidean distance, Manhattan distance, the hamming distance and cosine distance.. Below is the results table with different approaches and methods where

E = Euclidean Distance
M = Manhattan Distance
H= Hamming Distance
C = Cosine Distance
MM = MinMaxScalar()
SS = StandardScalar()

PCA	Over sample	Distance	K	Scalar	Recall	F1	Accuracy	RMSE
No	Yes	E	6	SS	0.70	0.80	**0.822**	0.421
No	Yes	M	7	SS	0.73	0.80	**0.815**	0.429
No	Yes	H	20	SS	0.71	76	**0.773**	0.476
No	Yes	C	6	SS	0.70	0.80	**0.822**	0.421
Yes	Yes	E	3	MM	0.78	0.83	**0.765**	0.484
Yes	Yes	M	5	MM	0.71	0.80	**0.731**	0.484
Yes	Yes	H	3	MM	0.74	0.70	**0.726**	0.477
Yes	Yes	C	3	MM	0.79	0.83	**0.765**	0.491
Yes	No	E	7	SS	0.89	0.85	**0.765**	0.469
Yes	No	M	12	SS	0.88	0.85	**0.772**	0.484
Yes	No	H	3	SS	1.00	0.86	**0.758**	0.454
Yes	No	C	11	SS	0.91	0.86	**0.779**	0.49
No	No	E	9	No	0.90	0.85	**0.765**	0.484
No	No	M	13	No	0.94	0.87	**0.793**	0.496
No	No	H	25	No	0.91	0.84	**0.752**	0.465
No	No	C	9	No	0.95	0.86	**0.765**	0.465

From the table we can say that the Euclidean distance and Cosine distance had almost the same accuracy and other values for corresponding approaches.We have achieved highest accuracy when we have used only sampled data without doing PCA , but the Recall and F1 are bit low for target 1. and very high for target 0. And when No PCA, No scaling was used the accuracy was moderate , The Recall and F1 score are very high for target 1 but bit low for target 0. When we have used both PCA and Scaling the results were decent with not very high nor very low accuracy.

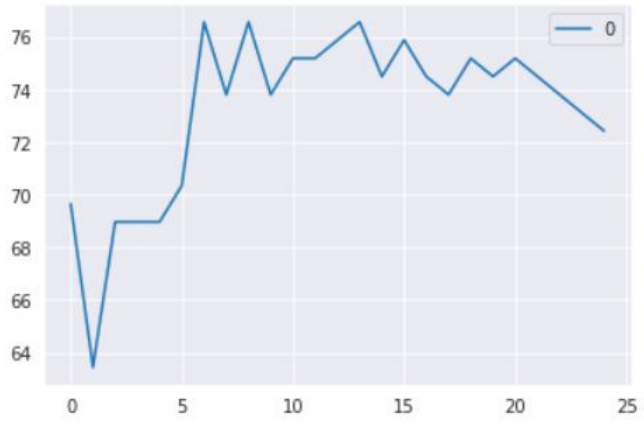


Figure 9 - Accuracy graph for KNN with K range 1-25

The above graph is a simple line to represent the changes in accuracy score as the K value changes. we can see that accuracy increased drastically after k =5 and remained high.

A small K value indicates low bias and high variance . As it increases variance decreases and bias increases [4]. As low K value may result in overfitting of data it is better to increase the value of k for better accuracy.

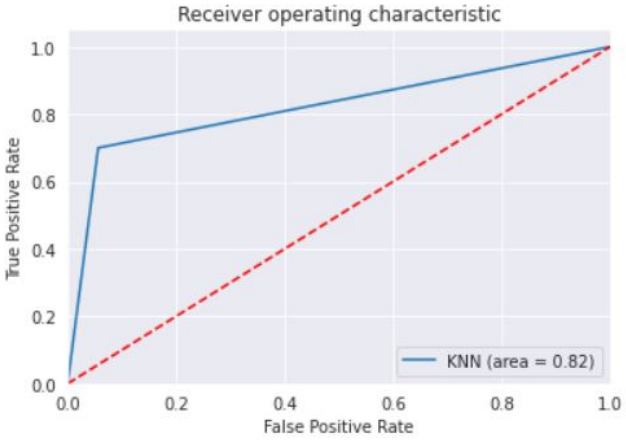


Figure 10 - KNN ROC Curve

The above graph is a sample ROC curve. We have used the AreaUnderCover (AUC) method to calculate the ROC curve.

Contributions:

Ali:

- Created a basic logistic regression classifier to start the project with some simple pre-processing methods such as handling missing values and binarizing categorical columns.
- Created and tested multiple random forest models. I used different methods to try to improve the results such as handling outliers, creating age groups and oversampling the minority data.
- Created a GridSearchCV function that can be used for our models to find the best parameters for our data.

Ragini:

- Created SVM hyperplane classifier to further improve the accuracy and F-Score with data pre-processing methods such as data cleaning, standardization using minimax scaler, removing duplicates and handling null values.
- As a first step of SVM model creation, I have used the baseline predictor for checking our metrics (accuracy, TPR, FPR) on that predictor.
- Created additional metric ROC Curve to check TPR and FPR and the performance of SVM.

Akshay:

- Build a KNN classifier with different parameters , preprocessing and normalisation techniques to achieve good accuracy
- Calculated RMSE which helps to determine the difference between the predicted values to the actual values . Lower the value more efficiently the model is.
- Plotted ROC curve and also calculated AUC

References

[1] https://www.kaggle.com/akhan890/liver-disease-lab-data

[2] https://www.kaggle.com/uciml/indian-liver-patient-records

[3] https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0

[4] https://mc.ai/chapter-1-k-nearest-neighbours-classifier/

[5] https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a

[6] https://medium.com/datadriveninvestor/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f

[7] Surname A and Surname B 2009 *Journal Name* **23** 544

[5] F. Markowetz. "Klassifikationmit support vector Machines". http://lectures.molgen.mpg.de/statistik03/docs/Kapitel16.pdf, 2003.

[6] W. W. Chapman, M.Fizman, B. E. Chapman, and P. J. Haug, "A Comparison of Classification Algorithms to Automatically Identify Chest X-Ray Reports That Support Pneumonia "journal of biomedical informatics,vol.34,pp 4: 14,2001.